

In the Claims:

Please amend the claims as indicated below:

1. (Currently amended) A system, comprising:

one or more processors; and

a memory storing program instructions, wherein the program instructions are
executable by the one or more processors to implement;

a virtual machine;

a plurality of subsystems configured to execute within ~~the~~ [[a]] virtual machine ~~on the system~~, wherein two or more of the plurality of subsystems ~~each~~ provide [[a]] different versions of an isomorphic interface to functions of the subsystems; and

a proxy mechanism configured to generate, for one of the plurality of subsystems at runtime of the one of the plurality of subsystems, a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems ~~that provides a correct version of the isomorphic interface for one of the plurality of subsystems at runtime of the one of the plurality of subsystems;~~

wherein the proxy is configured to:

receive a call to the isomorphic interface from the one of the plurality of subsystems;

convert the call in accordance with the version of the isomorphic interface

provided by the particular one of the two or more subsystems; and

forward the converted call to the particular one of the two or more subsystems for execution.

2. (Currently amended) The system as recited in claim 1, wherein the proxy mechanism is further configured to generate, for another one of the plurality of subsystems at runtime of the one of the plurality of subsystems, another proxy to a different version of the isomorphic interface provided by another one of the two or more subsystems ~~that provides a different version of the isomorphic interface for another one of the plurality of subsystems at runtime of the other one of the plurality of subsystems~~.

3. (Currently amended) The system as recited in claim 1, wherein, to generate a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems ~~that provides a correct version of the isomorphic interface for one of the plurality of subsystems at runtime of the one of the plurality of subsystems~~, the proxy mechanism is further configured to determine that the particular one of the two or more subsystems ~~that provides the a correct version of the isomorphic interface for the one of the plurality of subsystems at runtime~~.

4. (Currently amended) The system as recited in claim 3, wherein the one of the plurality of subsystems is configured to specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to determine that the particular one of the two or more subsystems ~~that provides the correct version of the isomorphic interface~~ according to said specification and to generate the proxy according to said specification.

5. (Currently amended) The system as recited in claim 1, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the particular one of the two or more subsystems to the one of the plurality of subsystems.

6. (Currently amended) The system as recited in claim 1, wherein the proxy mechanism is further configured to:

receive the call to the isomorphic interface from the one of the plurality of subsystems; and

generate [[a]] the proxy to the version of the isomorphic interface provided by the particular one of the two or more subsystems that provides a correct version of the isomorphic interface for one of the plurality of subsystems at runtime of the one of the plurality of subsystems in response to said call to the isomorphic interface from the one of the plurality of subsystems.

7. (Currently amended) The system as recited in claim 1, wherein the proxy mechanism is further configured to provide an API interface to the proxy mechanism for the plurality of subsystems, wherein the API interface is configured for use accessible by the subsystems to specify isomorphic interfaces provided by other ones of the subsystems to be proxied ~~to~~ by the proxy mechanism.

8. (Currently amended) The system as recited in claim 1, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems using Java Reflection.

9. (Original) The system as recited in claim 1, wherein the virtual machine is a Java Virtual Machine (JVM).

10. (Original) The system as recited in claim 1, wherein the one of the plurality of subsystems is an application, and wherein the two or more subsystems are versions of a runtime library.

11. (Original) The system as recited in claim 1, wherein the one of the plurality of subsystems and the two or more subsystems are applications.

12. (Original) The system as recited in claim 1, wherein the plurality of subsystems are mobile agents.

13. (Currently amended) A system, comprising:

one or more processors; and

a memory storing program instructions, wherein the program instructions are executable by the one or more processors to implement:

a virtual machine;

a plurality of subsystems configured to execute within ~~[[a]]~~ the virtual machine ~~on the system;~~ and

a proxy mechanism configured to:

generate a proxy for a version of an interface between two of the plurality of subsystems at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

generate another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the

interface.

14. (Original) The system as recited in claim 13, wherein the proxy is configured to:

receive from a first of the two subsystems a call to a second of the two subsystems;

convert the call in accordance with the version of the interface; and

forward the converted call to the second subsystem for execution by the second subsystem.

15. (Original) The system as recited in claim 14, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the second subsystem to the first subsystem.

16. (Original) The system as recited in claim 14, wherein the proxy mechanism is further configured to generate the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

17. (Original) The system as recited in claim 14, wherein the proxies are configured to convert the calls between the subsystems using Java Reflection.

18. (Original) The system as recited in claim 13, wherein a first of the two subsystems is configured to specify the interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to generate the proxy for the interface between the two subsystems in accordance with said specification.

19. (Original) The system as recited in claim 13, wherein the virtual machine is a Java Virtual Machine (JVM).

20. (Currently amended) A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a proxy mechanism configured to generate, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of ~~[[an]]~~ the isomorphic interface ~~for another subsystem at runtime of the other subsystem;~~

wherein the proxy is configured to:

receive a call to the isomorphic interface from the ~~other~~ subsystem;

convert the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems; and

forward the converted call to the particular one of the plurality of subsystems for execution.

21. (Currently amended) The system as recited in claim 20, wherein the proxy mechanism is further configured to generate, for a different subsystem at runtime of a different subsystem, another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface ~~for a different subsystem at runtime of the different subsystem.~~

22. (Currently amended) The system as recited in claim 20, wherein the proxy mechanism is further configured to generate another proxy configured to return

results of said execution from the particular one of the plurality of subsystems to the ~~other~~ subsystem.

23. (Currently amended) The system as recited in claim 20, wherein ~~other~~ the subsystem is configured to specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to generate the proxy according to said specification.

24. (Original) The system as recited in claim 20, wherein the program instructions are executable by the processor to implement a virtual machine in the memory, wherein the proxy mechanism, the subsystems, and the proxy are configured to operate within the virtual machine.

25. (Currently amended) The system as recited in claim [[20]] 24, wherein the virtual machine is a Java Virtual Machine (JVM).

26. (Currently amended) The system as recited in claim 20, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems using Java Reflection.

27. (Currently amended) A computer system, comprising:

means for generating proxies to versions of an isomorphic interface provided by two or more subsystems for one or more other subsystems in a virtual machine at runtime of the one or more other subsystems;

means for the proxies to convert calls from the one or more other subsystems to the versions of the isomorphic interface provided by the two or more subsystems; and

means for the proxies to forward the converted calls to the two or more subsystems for execution.

28. (Currently amended) A computer-implemented method, comprising:

a proxy mechanism generating, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of [[an]] the isomorphic interface at runtime of the subsystem;

the proxy receiving a call to the isomorphic interface from the subsystem;

the proxy converting the call in accordance with [[a]] the version of the isomorphic interface provided by the particular one of the plurality of subsystems; and

the proxy forwarding the converted call to the particular one of the plurality of subsystems for execution.

29. (Currently amended) The computer-implemented method as recited in claim 28, further comprising the proxy mechanism generating another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface for another subsystem at runtime of the other subsystem.

30. (Currently amended) The computer-implemented method as recited in claim 28, further comprising:

the proxy mechanism generating another proxy configured to return results of executing the call from the particular one of the plurality of subsystems to

the subsystem;

the particular one of the plurality of subsystems executing the call; and

the other proxy returning results of said executing the call to the subsystem.

31. (Currently amended) The computer-implemented method as recited in claim 28, further comprising the subsystem specifying the isomorphic interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems according to said specification.

32. (Currently amended) The computer-implemented method as recited in claim 28, further comprising the proxy mechanism receiving the call to the isomorphic interface, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems in response to said receiving the call.

33. (Currently amended) The computer-implemented method as recited in claim 28, wherein the proxy converts the call using Java Reflection.

34. (Currently amended) The computer-implemented method as recited in claim 28, wherein the proxy mechanism, the subsystems, and the proxy are implemented within a computer-implemented virtual machine ~~is a Java Virtual Machine (JVM).~~

35. (Currently amended) The computer-implemented method as recited in claim 28, wherein the subsystem is an application, and wherein the plurality of subsystems are versions of a runtime library.

36. (Currently amended) The computer-implemented method as recited in claim 28, wherein the subsystem and the plurality of subsystems are applications.

37. (Currently amended) The computer-implemented method as recited in

claim 28, wherein the subsystem and the plurality of subsystems are mobile agents.

38. (Currently amended) A computer-implemented method, comprising:

generating a proxy for a version of an interface between two of a plurality of subsystems in a virtual machine at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

generating another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

39. (Currently amended) The computer-implemented method as recited in claim 38, further comprising:

the proxy receiving from a first of the two subsystems a call to a second of the two subsystems;

the proxy converting the call in accordance with the version of the interface; and

the proxy forwarding the converted call to the second subsystem for execution by the second subsystem.

40. (Currently amended) The computer-implemented method as recited in claim 39, wherein the proxy converts the call in accordance with the version of the interface using Java Reflection.

41. (Currently amended) The computer-implemented method as recited in claim 39, further comprising generating another proxy configured to return results of said execution from the second subsystem to the first subsystem.

42. (Currently amended) The computer-implemented method as recited in claim 39, further comprising generating the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

43. (Currently amended) The computer-implemented method as recited in claim 38, further comprising:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

44. (Currently amended) The computer-implemented method as recited in claim 38, wherein the virtual machine is a Java Virtual Machine (JVM).

45. (Currently amended) A computer-accessible storage medium comprising program instructions, wherein the program instructions are ~~configured~~ computer-executable to implement:

a proxy mechanism generating, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of ~~[[an]]~~ the isomorphic interface ~~at runtime of the subsystem;~~

the proxy receiving a call to the isomorphic interface from the subsystem;

the proxy converting the call in accordance with [[a]] the version of the isomorphic interface provided by the particular one of the plurality of subsystems; and

the proxy forwarding the converted call to the particular one of the plurality of subsystems for execution.

46. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the program instructions are further ~~configured~~ computer-executable to implement the proxy mechanism generating another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface for another subsystem at runtime of the other subsystem.

47. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the program instructions are further ~~configured~~ computer-executable to implement:

the proxy mechanism generating another proxy configured to return results of executing the call from the particular one of the plurality of subsystems to the subsystem;

the particular one of the plurality of subsystems executing the call; and

the other proxy returning results of said executing the call to the subsystem.

48. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the program instructions are further ~~configured~~ computer-executable to implement the subsystem specifying the isomorphic interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems according to said specification.

49. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the program instructions are further ~~configured~~ computer-executable to implement the proxy mechanism receiving the call to the isomorphic interface, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems in response to said receiving the call.

50. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the proxy converts the call using Java Reflection.

51. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the proxy mechanism, the subsystems, and the proxy are implemented within a virtual machine-is a Java Virtual Machine (JVM).

52. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the subsystem is an application, and wherein the plurality of subsystems are versions of a runtime library.

53. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the subsystem and the plurality of subsystems are applications.

54. (Currently amended) The computer-accessible storage medium as recited in claim 45, wherein the subsystem and the plurality of subsystems are mobile agents.

55. (Currently amended) A computer-accessible storage medium comprising program instructions, wherein the program instructions are ~~configured~~ computer-executable to implement:

generating a proxy for a version of an interface between two of a plurality of subsystems in a virtual machine at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the

two subsystems in accordance with the version of the interface; and

generating another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

56. (Currently amended) The computer-accessible storage medium as recited in claim 55, wherein the program instructions are further ~~configured~~ computer-executable to implement:

the proxy receiving from a first of the two subsystems a call to a second of the two subsystems;

the proxy converting the call in accordance with the version of the interface; and

the proxy forwarding the converted call to the second subsystem for execution by the second subsystem.

57. (Currently amended) The computer-accessible storage medium as recited in claim 56, wherein the proxy converts the call in accordance with the version of the interface using Java Reflection.

58. (Currently amended) The computer-accessible storage medium as recited in claim 56, wherein the program instructions are further ~~configured~~ computer-executable to implement generating another proxy configured to return results of said execution from the second subsystem to the first subsystem.

59. (Currently amended) The computer-accessible storage medium as recited in claim 56, wherein the program instructions are further ~~configured~~ computer-executable

to implement generating the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

60. (Currently amended) The computer-accessible storage medium as recited in claim 55, wherein the program instructions are further ~~configured~~ computer-executable to implement:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

61. (Currently amended) The computer-accessible storage medium as recited in claim 55, wherein the virtual machine is a Java Virtual Machine (JVM).